

REMARKS

Claims 1, 9-11, 20, 21, 23, and 24 are pending in the present application. Claims 2, 3, 5, 7, 13, 14, 16, 18, 22, and 25-30 are canceled. Claims 1, 11, 12, 23, and 24 are amended. Reconsideration of the claims is respectfully requested.

I. Examiner Interview

Applicant thanks the Examiner for the courtesies extended in the telephone interview on August 16, 2005. During the interview, Applicant discussed various distinctions between the applied prior art and the present invention. Applicant presented proposed amendments on August 18, 2005. During a telephone conversation on August 22, 2005, the Examiner agreed that the proposed amendments, which are officially presented above, overcome the rejections of record.

II. 35 U.S.C. § 103, Alleged Obviousness

The Office Action rejects claims 1-3, 5, 7, 9-14, 16, 18, 20, 21-26, 28, and 29 under 35 U.S.C. § 103 as being unpatentable over "The Reflective State Pattern" by *Ferreira et al.* This rejection is respectfully traversed.

Independent claim 1 is amended to incorporate the limitations of 2, 3, 5, and 7. *Ferreira* teaches methods for creating states and transitions; however, these methods are part of the FSMController (see pages 4-6). *Ferreira* also teaches that states and transitions are created metaobjects (see page 9).

However, *Ferreira* does not teach an initializer object that includes a table element array creation method and a table variable array creation method, wherein the state machine object is configured to call the table element array creation method to create a table object using the results of the table element array creation method and wherein the state machine object is configured to call the table variable array creation method to create an array of state variables using the results of the table variable array creation method. It follows that *Ferreira* does not teach a table object that is configured to create an array of state transition objects based on the array of state variables and return the array of state transition objects to the state machine object.

Claim 1 presents a specific manner in which the array of state transition objects is created. Claim 1 does not merely use generally known constructor methods to create the transitions. In fact, *Ferreira* teaches away from the invention of claim 1, because *Ferreira* teaches using create methods in the FSMController itself to create states and transitions from metaobjects that define states and transition conditions for a software state machine, rather than having the FSMController call methods in an initializer object.

Independent claims 12 and 23 recite subject matter addressed above with respect to claim 1 and are allowable for similar reasons. Since claims 9, 10, 20, and 21 depend from claims 1 and 12, the same distinctions between *Ferreira* and the invention recited in claims 1 and 12 apply for these claims.

Independent claim 11 is amended to incorporate the limitations of claims 25-27. *Ferreira* teaches that FSMState has a handleEvt() method and FSMTransition has a handleTransition() method. Any actions or events that result from a state transition must be specifically coded into these methods.

In contradistinction, the invention recited in claim 11 provides a set of graphical user interfaces for entry of a plurality of states, one or more actions, one or more inputs, one or more conditions, one or more events, one or more triggers, and a plurality of state transitions for a software state machine. The state machine object is configured to use the initializer object to create an array of state transition objects based on the plurality of state transitions, wherein each state transition object in the array of state transition objects references at least one of the one or more conditions that causes a given state transition in the software state machine. Also, a first state transition object in the array of state transition objects defines an action to take responsive to a given state transition and a second state transition object in the array of state transition objects defines an event to be generated responsive to a given state transition.

As stated above, in *Ferreira*, actions and events that occur as a result of a state transition must be specifically programmed into the handleTransition() method. Therefore, *Ferreira* does not render claim 11 obvious, because *Ferreira* does not teach or fairly suggest each and every claim limitation. Also, a person of ordinary skill in the art would not find it obvious to modify *Ferreira* to arrive at the presently claimed invention. Independent claim 24 recites subject matter addressed above with respect to claim 11 and is allowable for similar reasons.

Therefore, Applicant respectfully requests withdrawal of the rejection of claims 1, 9, 10, 11, 12, 20, 21, 23, and 24 under 35 U.S.C. § 103.

The Office Action rejects claims 27 and 30 under 35 U.S.C. § 103 as being unpatentable over "The Reflective State Pattern" by *Ferreira et al.* in view of *Nguyen* (U.S. Patent No. 6,751,753). This rejection is respectfully traversed.

Independent claims 11 and 24 are amended to incorporate features previously presented in claims 27 and 30. Claims 11 and 24 are also believed to be allowable over the proposed combination of *Ferreira* and *Nguyen*. More specifically, *Nguyen* does not make up for the deficiencies of *Ferreira*. That is, *Nguyen* does generally teach graphical user interfaces that allow a user to specify state machine modifications; however, *Nguyen* does not teach or suggest a set of graphical user interfaces for entry of a plurality of states, one or more actions, one or more inputs, one or more conditions, one or more events, one or more triggers, and a plurality of state transitions for a software state machine. *Nguyen* also fails to teach or suggest a state machine object that is configured to use an initializer object to create an array of state transition objects based on the plurality of state transitions, wherein each state transition object in the array of state transition objects references at least one of the one or more conditions that causes a given state transition in the software state machine. In addition, *Nguyen* fails to teach a first state transition object in the array of state transition objects defines an action to take responsive to a given state transition and a second state transition object in the array of state transition objects defines an event to be generated responsive to a given state transition.

Therefore, *Ferreira* and *Nguyen*, taken individually or in combination, fail to render claims 11 and 24 obvious. Applicant respectfully submits that the rejection of claims 27 and 30 under 35 U.S.C. § 103 does not apply to current claims 11 and 24.

III. Conclusion

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Aug. 23, 2005

Respectfully submitted,



Stephen R. Tkacs
Reg. No. 46,430
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicant